2

Towards Intelligent Automated Forces For Simnet

Semi-Annual Report to the Office of Naval Research
for the Period 4/1/91 - 9/30/91
on Contract Number N00014-91-J-1624

Paul S. Rosenbloom, PI
Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292

October 15, 1991

One of our long-term research goals is to construct Soar-based autonomous intelligent agents that can act as independent players in multi-agent simulation environments. The Soar architecture has been designed with the goal of being able to support general intelligence, and various systems built within Soar have demonstrated a range of planning, learning, language, and robotics behaviors. However, there is not yet a single agent that combines the architecture with sufficient knowledge to allow it to survive and behave appropriately in complex simulated environments. The focus of this contract is on performing seed research towards this end.

The work is being pursued in the context of a simplified battlefield environment – called *GridWorld*, and developed at the Hughes AI Center – which provides a two-dimensional, multi-agent, real-time world in which both space and time are represented as continuous quantities. During the initial segment of this reporting period we focused on infrastructure issues: converting Gridworld to run in our environment, enhancing it in essential ways, and interfacing Soar to it. Once this was accomplished, we began the task of creating a Soar-based agent with a broad, but relatively shallow, approach. The goal was to get an agent that could behave in a reactive fashion to various contingencies, without yet worrying about more sophisticated cognitive activities such as planning and learning. We are now looking at integrating such higher-level behaviors into the agent. One particular area we are examining is navigation and spatial reasoning.

The following sections provide additional detail on our progress with these infrastructure and agent-construction issues.

Converted And Enhanced the Gridworld Simulator

As we received it, the GridWorld simulator ran on an Apple Macintosh, so the first task was to convert it to the Sun and Unix, using X-Windows for the graphics (via Sun's Lispviews). New graphics capabilities were added, including a global picture of the battlefield (instead of just a tank's point of view). Bottlenecks were identified and removed resulting in a three-fold improvement in processing time. Other enhancements included an interactive capability for stepping through the simulation, and a number of improved algorithms for line-of-sight calculations and more realistic directional sensing.

Interfaced Soar to the Gridworld Simulator

A simulated tank in GridWorld receives perception and effects commands via a user-written control procedure which is called periodically in the simulation loop. Interfacing GridWorld and Soar consisted in writing an I/O module which converted these perception inputs and command outputs to/from Soar's working memory. This involved some choices as to how much detail should be filtered out by the I/O module and in what form the information should be presented to Soar's working memory. The general principle followed was that arithmetic quantities and calculations reside in the I/O module and only symbolic information should reside in Soar's working memory.

Constructed Agents In Soar

We have constructed a succession of more sophisticated GridWorld tank controllers in Soar. The actual control of a tank was accomplished by writing Soar productions. Emphasis was first placed on programming low-level behaviors which enabled the tank to successfully move from a start position to a goal position, avoiding obstacles and other tanks along the way. Initially, the tank was purely reactive in that it had no internal representation of the world and hence no plan to follow; instead, its behaviors were essentially mappings from its perception and state to command outputs. The behaviors included turning and moving towards ends of visible walls, avoiding enemy tanks and following a fixed distance behind a friendly leader tank, shooting at visible enemy tanks, avoiding obstacles sensed at close range, and moving towards its goal. Because these behaviors can sometimes conflict, arbitration was needed to choose between them: this was accomplished by Soar's preference scheme and support for interruption and resumption of operators. A model of behaviors as continuous operators which persist over time was developed and led to new ways to regard Soar's decision cycle.

This simple approach enabled the tank to accomplish some goals in (simulated) real time, but the result was a very primitive form of "intelligence". One aspect of this was a lack of navigational capabilities – the tank had no representation of a global map, and hence no ability to use such a map to plan efficient routes. This led to two investigations: how to enable the agent to learn a map from its reactive exploration, and how to then use this map to navigate more efficiently in a dynamic world where the map cannot be guaranteed to be accurate. The first aim was accomplished by having the agent deliberately remember locations where it perceived landmarks and took actions. These perception-action procedures were then massaged (by the agent) into a procedural map which was then followed in a more efficient manner because many of the subgoal decisions were eliminated and some of the paths traversed were shortened. Since this map abstracted out local obstacle avoidance motions, and those behaviors were still active, the agent was still able to navigate in a dynamic environment while following the map. Most recently, a more expressive map representation was investigated which allowed the agent to answer self-queries concerning routes in the terrain via an internal simulation. This internal simulation paves the way for more complete planning capabilities in reasoning about the problem of moving around a dynamic uncertain world.

The Next Steps

During the next period we expect to focus on four key issues with respect to navigation and its integration with the overall agent. The first issue is the types of tasks to be performed in the environment that interact with the agent's navigational abilities. If the sole task is to get from point A to point B, nothing more than a simple algorithmic route planner may be required. However, if tasks require deciding what points to move between, and perhaps altering them in progress, the agent may itself need to understand more about the spatial-temporal structure of its world and routes. The second issue is the ranges of navigation-related information that will be available in performing these tasks; for example, maps of various resolution and quality, absolute coordinate information and perceptual information. A robust navigation facility would adapt to whatever information is (or is not) available. The third issue is the types of external tools available for use by the agent. There is no a priori reason to require that all navigational processing be done in the head; on the contrary, if the agent can use sophisticated calculational and planning tools, its performance may be significantly enhanced. The fourth issue is the types of internal structures and processes the agent must have to perform the requisite tasks given the available information and tools. This is the core of how the agent navigates, and integrates its navigational abilities with its other abilities, such as to plan, react, take instructions, and learn. In investigating all four of these issues we will, as much as possible, utilize the rather extensive range of results, algorithms and tools that have recently been developed recently in the areas of spatial reasoning and robotic navigation.

Statement A per telecon
Dr. Teresa McMullin ONR/Code 1133
Arlington, VA 22217-5000

NWW 11/18/91